The Pillars of Skill-Acquisition and Generalization; Why efficient General Intelligence requires Multi-Component Integration

Felix Steinbauer*
School of Computation, Information and Technology (CIT)
Munich Center for Machine Learning (MCML)
Technical University of Munich
Munich, Germany
felix.steinbauer@tum.de

Bardh Prenkaj
School of Social Sciences and Technology
Munich Center for Machine Learning (MCML)
Technical University of Munich
Munich, Germany

Efstratios Zaradoukas
CIT
Munich Center for Machine Learning (MCML)
Technical University of Munich
Munich, Germany

Mrigyen Sawant
CIT
Technical University of Munich
Munich, Germany

Florian Kofler
Department of Quantitative Biomedicine
University of Zurich
Switzerland

Gjergji Kasneci
CIT
Munich Center for Machine Learning (MCML)
Technical University of Munich
Munich, Germany

Abstract

Breakthroughs of Large Language Models (LLMs) have rekindled hopes for broadly capable artificial intelligence (i.e., Artificial General Intelligence (AGI)). Yet, these models still exhibit notable limitations – particularly in deductive reasoning and *efficient* skill acquisition. In contrast, *neuro-symbolic* approaches can exhibit more robust generalization across diverse tasks, as they integrate sub-symbolic pattern extraction with explicit logical structures. In this position paper, we go *a step further* and dissect generalizing systems into *six* pillars: well-defined model specificity, (human) capability encoding, dynamic knowledge acquisition & transfer, meaningful representations, abstraction & hierarchies, as well as the synergy effects resulting from component interactions. Based on historical and contemporary Artificial Intelligence (AI) approaches, we conclude that such *a multi-component implementation strategy is necessary for efficient general intelligence*. Our position is reinforced by the latest performance gains on the Abstraction and Reasoning Corpus (ARC) generalization benchmark.

*Corresponding author: felix.steinbauer@tum.de

1 Introduction

Across decades of progress, research on "artificial intelligence" has often centered on narrow tasks and small leaps in computational automation, without necessarily pursuing robust, human-like intelligence. This changed with the rise of large neural networks – models that excel in pattern extraction and display intriguing emergent capacities [Bubeck et al., 2023]. Yet, while these black-box approaches are remarkable in many respects, they also suffer from opaque decision-making processes and often exhibit only *local* forms of generalization. They thus provide limited insights into the core mechanisms underlying *flexible, human-level* intelligence.

Motivated by these gaps, an increasing number of researchers suggest incorporating symbolic reasoning into deep learning pipelines, giving rise to *neuro-symbolic* approaches [d'Avila Garcez and Lamb, 2023, Keber et al., 2024]. By preserving the neural model's strengths in statistical pattern recognition and combining them with symbolic structures that allow for compositional logic, explainable decisions, and interpretability, neuro-symbolic methods promise broader skill-acquisition efficiency, deeper semantic understanding, and safer real-world deployment [Hernández-Orallo, 2020, Hassija et al., 2024].

However, merely layering symbolic modules on top of neural back-ends does not automatically confer *general* intelligence. To foster meaningful progress, we must first **define** (1) the gist of (artificial) intelligence, especially in terms of *skill acquisition efficiency*, then pinpoint how best to **evaluate** (1) a model's capacity to abstract knowledge from sparse data and adapt to novel tasks. On this basis, we evaluate predominant research directions, such as Large Reasoning Models (LRMs) and neuro-symbolic approaches (2). We derive our position that **a novel, multi-component implementation strategy is necessary for efficient general intelligence**. Our core contribution is the identification of six different fundamental pillars for achieving efficient generalization (3); (3.1) Model Specificity, (3.2) (Human) Capability Encoding, (3.3) Meaningful Representations, (3.5) Knowledge Acquisition & Transfer, (3.4) Abstractions & Hierarchies, and (3.6) Multi-Component Synergy. We conclude with implications for future research directions and practical system design (4).

Defining Intelligence as Skill-Acquisition Efficiency Despite centuries of study, intelligence remains notoriously difficult to define comprehensively [Legg and Hutter, 2007]. We adopt the formulation by Chollet [2019] that views the intelligence of a system as "*a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty*." This perspective shifts attention from raw *performance* on a single task to the *ability to learn new* tasks under constraints – such as limited data, novel transformations, or minimal prior knowledge.

An agent that extracts greater competence (skills, insights, etc.) from identical training conditions is inherently more efficient at acquiring skills. In evolution, this efficiency holds an inherent advantage. Most modern intelligence tests are based on the related concepts of fluid and crystallized intelligence. Figure 3 from [Chollet, 2019] visualizes this *information conversion ratio* from situational to operational space nicely. All other factors (e.g., prior knowledge, curriculum size, development efforts (e.g., inductive bias), training time & strategy, intrinsic task difficulty) should be *controlled for* to isolate the system's skill-acquisition efficiency. Even the competence a developer (or the development process) (un)consciously puts into the model (e.g., by hyperparameter choice) shall be accounted for to get a clean measure of the system's own skill-acquisition efficiency. For more on "developer-aware" generalization see Appendix B.

Of course, this intelligence definition has shortcomings but we believe[2] that it is currently the most *complete* and *correct* notion of intelligence we have, which is applicable to humans as well as machines. A consequence of this perspective is that **skill-acquisition efficiency** is at the heart of what sets "general" intelligence apart from specialized or over-engineered solutions [Bober-Irizar and Banerjee, 2024]. Hence, if the field's ambition is true *general* intelligence – rather than a proliferation of specialized or heavily handcrafted solutions – then adopting metrics and methods highlighting *skill-acquisition efficiency* becomes indispensable. This, in turn, requires reliable ways to evaluate how well a model performs under low-data, unseen, or compositional scenarios – where brute-force training or naive memorization is infeasible.

[2]For more detailed arguments, please see chapters I and II of Chollet [2019]

Benchmarking for Generality Skill-acquisition efficiency, i.e., the amount of competence gained from a fixed amount of data or experience, should be evaluated independently of a system's final performance. The prior knowledge of a system matters as it gives a head start on performance. Traditional benchmarks often conflate performance with the data or developer-engineered interventions needed to achieve it. Here is where the ARC comes in; it tries to isolate and measure a system's inherent learning efficiency [Chollet, 2019]. The ARC benchmarks (currently in version ARC-AGI-1 and ARC-AGI-2) consist of small, diverse puzzles that test "core knowledge" concepts like spatial manipulation, color/object transformations, or compositional logic [Moskvichev et al., 2023]. Besides their simple format/setting, the crucial challenge is that no task is like the other, and the test set contains tasks that are unseen during training. The point of ARC is to force the model to learn the least common denominator of all scenarios, meaning that it needs to generalize over and flexibly operate on the underlying geometric concepts. This requires some form of generalization capability over a set of problems, making the benchmark robust against performance saturation by large, curated datasets.

Despite being straightforward for humans, ARC tasks have proven unexpectedly difficult for computational models, with only about half the tasks consistently solved on the private ARC-AGI-1 test set [Bober-Irizar and Banerjee, 2024, ARC Prize, 2024]. This difficulty emerges precisely because ARC demands *abstract generalization* over a minimal set of examples, thwarting superficial shortcuts. While ARC is not a perfect proxy for all human-level reasoning, it remains a valuable gauge of small-data adaptability, creative knowledge transfer, and flexible problem solving. The many caveats of ARC-AGI-1 were partially resolved in ARC-AGI-2 [Chollet, 2019, ARC Prize, 2025c]. As this second iteration exists only for a few months and not many custom approaches for this iteration exist yet, we will primarily focus on ARC-AGI-1 in this paper. Nevertheless, the setting and therefore the conclusions are similar for ARC-AGI-2. In what follows, we leverage ARC as a testbed to motivate why **efficient generalization requires multi-component integration**. For more details on limits, alternatives and why we still choose ARC, see Appendix D.

2 Alternative Views

To the best of our knowledge, there is currently no real alternative view focused on *efficient* generalization. However, the next closest approaches might be **(a) Extended LLMs** (2.1) as a not-so-efficient but generalizing method, and **(b) Neuro-Symbolic** methods (2.2) as an efficient but not generality-focused approach.

2.1 Extended Large Language Models

Transformers and LLMs have undeniably exhibited broad emergent capabilities, including surprising generalization and few-shot reasoning, across multiple domains [Bubeck et al., 2023, Webb et al., 2023]. When *extended* with techniques like chain-of-thought prompting and test-time fine-tuning, they can perform competitively even on ARC [Greenblatt, 2024a, Berman, 2024, ARC Prize, 2025a]. With resource-heavy test-time optimization, models like GPT-4 and Sonnet 3.5 achieve up to 87.5% on ARC-AGI-1, leading many to view LLMs as the foundation for future general-purpose AI[ARC Prize, 2025a].

Strengths of LLMs. Modern LLMs exhibit several key strengths. **Pre-training on Massive Corpora** allows for extensive self-supervised learning on diverse text sources. This way, LLMs acquire a wealth of representations, effectively consolidating and covering wide-ranging knowledge [Bubeck et al., 2023]. **Flexible Transfer of Knowledge** can be applied to handle various downstream tasks (including non-linguistic tasks expressed in language) with minimal fine-tuning, thanks to in-context learning, powerful embedding spaces, and diverse test-time strategies [Dong et al., 2023, Berman, 2024]. **Emergent Reasoning Behaviors** can be elicited through prompting strategies such as chain-of-thought or retrieval augmented generation. Such reasoning-like procedures within LLMs often improve the performance on complex tasks [Webb et al., 2023].

Challenges and Limitations. Despite impressive benchmark results, LLMs still exhibit significant hurdles regarding *efficient* generalization:

1. **Opaque and Brittle Emergence:** The extent to which LLMs can perform genuine abstract reasoning (versus pattern matching) remains an open debate [Valmeekam et al., 2023, Kaddour et al., 2023, Dziri et al., 2023, Lewis and Mitchell, 2024, Wang et al., 2024a, Lotfi et al., 2024, Schuurmans et al., 2024, ARC Prize, 2025f]. Their "emergent" abilities can be unreliable, hard to interpret, and domain-specific [Bober-Irizar and Banerjee, 2024]. For example, on ARC-AGI-1, the best performing LLMs achieve up to 87.5% but on ARC-AGI-2 only 4%, while humans consistently reach 100% [ARC Prize, 2025d].
2. **Data-Hungry and Costly:** Training large-scale transformers demands massive, human-generated corpora – and some fear we are reaching the upper limit of high-quality data for further scaling this approach [Sutskever, 2024]. In addition, fine-tuning and extensive resource-intensive test-time synthesis methods are expensive (money and time) [Sachdeva et al., 2024, Greenblatt, 2024a, Berman, 2024]. For example, on ARC-AGI-1, the amount of solved tasks **scales logarithmically** with the inference compute at test time [ARC Prize, 2025a]. For o3 to reach 75.7% on the semi-private ARC-AGI-1, $\$20$ and 13.8 minutes per task were necessary. To obtain 87.5%, roughly $\$3400$ and 3.7 hours[3] were reported [ARC Prize, 2025e]. In comparison, an average human STEM graduate reaches 98% requiring $\$10$ [ARC Prize, 2025d]. For further discussion on scaling efficiency, see Appendix H.3.
3. **Developer vs. Model Intelligence:** Many LLM-based successes rely heavily on *engineered prompting* and human-coded heuristics. The latest ARC-AGI-1 results reveal that while LLM-based approaches can outperform other methods on the public benchmark, they do so through *massive prompt engineering* [Greenblatt, 2024a, Berman, 2024]. Thus, high-level performance may reflect *developer-centric*[4] skill more than an intrinsic model capacity for generalization [Chollet, 2019, Dong et al., 2023, Yu et al., 2023, Bober-Irizar and Banerjee, 2024].
4. **Lack of Transparency:** Unlike modular designs, LLMs encode reasoning steps in vast weight matrices, limiting interpretability. This black-box nature impedes deeper analysis

[3] Corrected for 173 times more compute compared to *low* setting

[4] Definition see Appendix B

of the reasoning process and complicates improvements targeted at genuine compositional intelligence [Garcez and Lamb, 2023]. For more details, see Appendix A.

Conclusion for LLMs. Even though LLMs are powerful in practice, they do not generalize efficiently. Mahowald et al. [2024] draw parallels to the human brain's specialized "language areas," cautioning that forcing a language-dominant model to cover abstract non-linguistic tasks may be fundamentally inefficient. We cannot know yet if the continuing trend of extending purely neural LLMs with fancy strategies like Chain of Thought (CoT), In-Context Learning (ICL), Retrieval Augmented Generation (RAG), test-time compute, etc., will make them ultimately more efficient at generalization. This inefficiency might very well be fundamental, but we nevertheless encourage the continuing efforts to embed LLMs into neuro-symbolic frameworks. For a more detailed discussion, see Appendix H.3 and C.

2.2 Neuro-Symbolic Strategies

The term *neuro-symbolic* (sometimes abbreviated *NeSy*) can encompass a wide variety of hybrid architectures and learning strategies. While the specific mechanisms vary, the core idea is to marry *symbolic structures* (e.g., logic programs, Domain-Specific Languages (DSLs), knowledge graphs) with *neural components* (e.g., deep networks or learned embeddings) [Hitzler et al., 2022, Garcez and Lamb, 2023, Keber et al., 2024]. These two paradigms clearly complement each other [Bober-Irizar and Banerjee, 2024], already hinting at the potential of neuro-symbolic methods to tackle a broader range of tasks than each paradigm alone [Bober-Irizar and Banerjee, 2024, Chollet et al., 2025].

As multiple works have already surveyed the general advantages and disadvantages of neuro-symbolic approaches in depth [Hamilton et al., 2022, Hitzler et al., 2022, Garcez and Lamb, 2023, Keber et al., 2024, Bhuyan et al., 2024], we will not reiterate these existing arguments. Instead, we focus here on the key aspects *relevant for generalization*. Nevertheless, for completeness, we provide a brief discussion of symbolic approaches in Appendix G.

Relevance in LLMs Methods like chain-of-thought prompting and structured reasoning graphs already incorporate neuro-symbolic principles [Hitzler et al., 2022]. These techniques wrap neural transformers in symbolic scaffolding [Yu et al., 2023], improving performance across tasks. Examples include tree-of-thought [Yao et al., 2023] and graph-based reasoning [Besta et al., 2024]. Xu et al. [2024] demonstrate how such logical orchestration around LLM calls enhances reliability on diverse tasks. Consequently, also on ARC-AGI-1, the top LLM-based approaches incorporate symbolic heuristics to stabilize generalization [Franzen et al., 2024, Barbadillo, 2024, Chollet et al., 2025].

Advantages and Successes In recent years, researchers have increasingly aimed to harness the advantages of combining neural and symbolic paradigms[Hitzler et al., 2022, Garcez and Lamb, 2023]. That this strategy is fruitful could be shown by some first successes like the neuro-symbolic theorem prover AlphaGeometry [Trinh et al., 2024]. Also, the most recent ARC-AGI-1 findings [Chollet et al., 2025] show that neuro-symbolic approaches are a promising route to generalization. A clear illustration is Bober-Irizar and Banerjee [2024], who build upon a DSL-based ARC solver by adding learnable "concept formation" components, significantly boosting efficiency and success rates. Hybrid models can learn abstract concepts more compactly, leveraging both (i) a neural module to handle noisy or high-dimensional inputs and (ii) a symbolic module to enforce logical coherence and compositional reasoning. This synergy is particularly relevant in low-data tasks like ARC, where purely neural systems often overfit, and purely symbolic systems lack robust inductive priors. Table 3 in Appendix F summarizes a few representative state-of-the-art neuro-symbolic approaches that have been shown to be effective for generalization in ARC-like tasks.

Challenges and Limitations While recent work has demonstrated promising gains on ARC [Moskvichev et al., 2023, Chollet et al., 2025, Bober-Irizar and Banerjee, 2024], open challenges remain – most notably:

1. **Exploding Search Spaces.** Combining symbolic search with neural heuristics can mitigate the worst-case combinatorial complexity explosion, but designing these heuristics remains nontrivial [Bober-Irizar and Banerjee, 2024].
2. **Balancing Data Efficiency and Model Complexity.** For example, ARC tasks demand strong reasoning from minimal examples, stressing the importance of balanced architectures that do not over-parameterize [Moskvichev et al., 2023].

3. **Dynamic Concept Formation.** Handling ever-evolving domains requires neuro-symbolic methods that can *learn new concepts dynamically* rather than rely solely on a hard-coded DSL [Bober-Irizar and Banerjee, 2024].
4. **Underspecified methodology.** To the best of our knowledge, there is very little standardized methodology in the field of neuro-symbolic AI research. We are still searching for a consensus on the most effective ways to combine neural and symbolic approaches [Feldstein et al., 2024].
5. **Limited Focus on Generalization.** The promise of neuro-symbolic integration is as significant as it is unspecific. Combining the two major AI research paradigms of the last 80 years will open many possibilities; efficient generalization might be one of them, but not the prime focus of the strategy [Garcez and Lamb, 2023, Bhuyan et al., 2024].

Conclusion for Neuro-Symbolic Approaches Though the obstacles mentioned above are significant, the ability of neuro-symbolic methods to unify inductive and deductive reasoning is an especially potent strength – analogous to "System 1" vs. "System 2" thinking in human cognition [Kahneman, 2011, Garcez and Lamb, 2023]. As computational and data constraints grow more relevant, this marriage of neural and symbolic approaches will likely become unavoidable for efficient models. Unfortunately, the field of neuro-symbolic AI research is still in its infancy, quite underspecified regarding concrete methodology and not particularly focused on generalization, rendering it not particularly helpful for advances on skill-acquisition efficiency [Garcez and Lamb, 2023, Feldstein et al., 2024, Bhuyan et al., 2024].

2.3 Closing Remark

In summary, LLM-focused approaches can demonstrate remarkable capabilities but often rely on extensive engineering, computational resources, and data, with limited inherent interpretability and skill-acquisition efficiency. Neuro-symbolic methods stand at the intersection of statistical learning and explicit symbolic reasoning, promising synergy effects far beyond what either paradigm can achieve alone. They might be the most promising previously known route to efficient, transparent generalization. However, they only *hint* at the power of multi-concept integration and are underspecific when it comes to efficient generalization.

3 Pillars of Efficient Generalization

Based on historical examples, recent approaches, and models designed for the ARC benchmark, we propose that the following **fundamental pillars** are indispensable for attaining *efficient* generalization. A key insight of our work is the interrelation between these pillars, which we highlight by referencing connections between pillars using section references. For an overview, see Figure 1.

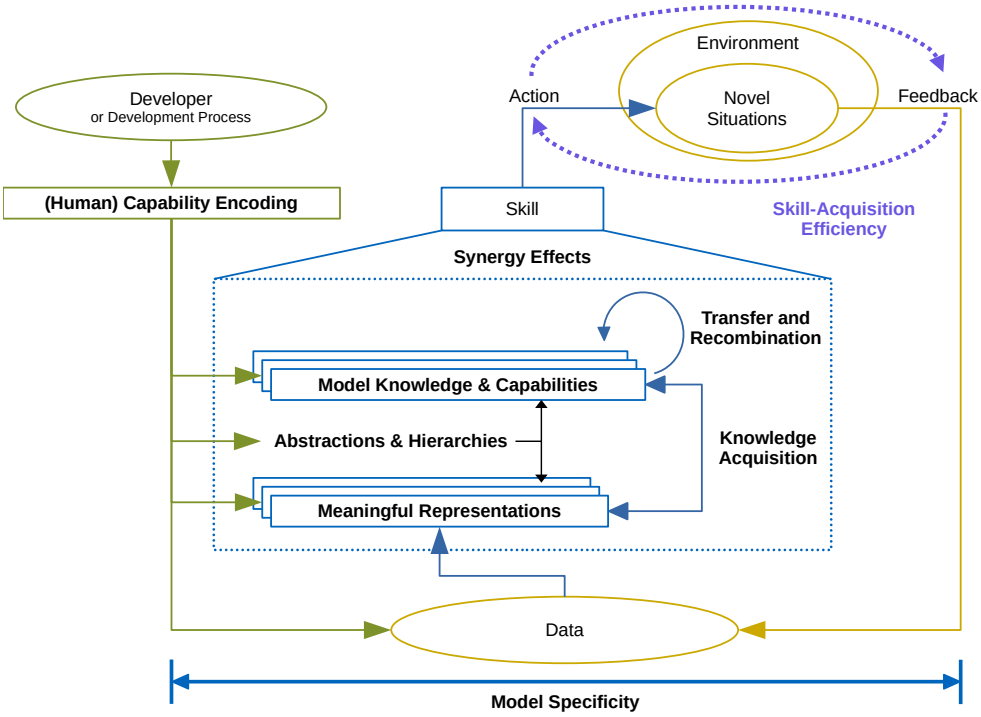

Figure 1: **Conceptual illustration of the six pillars for efficient generalization. Blue**: Components of a generalizing model. **Green**: Human development components. **Yellow**: Environment and Data. **Oval shapes**: Entities. **Bold**: Pillars (**Knowledge Acquisition** and **Transfer** separated for clarity). **Explanation**: The overall scope of the model, **data**, **development** is dictated by the **Model Specificity** (3.1). The **Developer** (or development process) is responsible for encoding **(Human) Capabilities** (3.2) into the model as well as preselecting data. The data is represented using **Meaningful Representations** (3.3). The model uses **Abstractions & Hierarchies** (3.4) to work at appropriate levels of granularity. On this basis, the model **extracts knowledge and competences** (3.5) which are reiterated over (**transfer** and **recombination**). Finally, all components of the model and the development process create **Multi-Component Synergy Effects** (3.6) that contribute to the model's **skill** level. The skill determines the **action** the model takes in a **novel situation** (which is determined by the **environment**). The situation will return some **feedback** signal, which is interpreted as new data. Based on this environment reaction, the model updates its inner states, improves its skill and (hopefully) generalizes to *the underlying task*. The (interactive) cycle between model action, environment reaction, model update and *skill improvement* is where we observe **skill-acquisition efficiency** in the end.

3.1 Model Specificity

Currently, we are unable to produce *efficient* models for general intelligence. For example, extended LLMs (see Section 2.1) generalize over a broad range of tasks but possess low *skill-acquisition efficiency* when accounting for all relevant factors (see Section 1). When hitting the practical limits of transformer scalability, the research community is forced to increase skill-acquisition efficiency to further improve generalization. The latest developments regarding reasoning models underpin this observation [DeepSeek-AI et al., 2025, Ballon et al., 2025].

We argue that the process should be the other way around: first achieving **high** skill-acquisition efficiency over a **limited scope**, which later needs to be extended, instead of aiming for **low** skill-acquisition efficiency over a **broad scope**, and later increasing the skill-acquisition efficiency. Consequently, a pillar for efficient generalization is properly fitting the model to the given scope. There are many steps in between *narrow task-specific AI* and *open-ended AGI*. We advocate for climbing this ladder slowly but thoroughly, aiming for high skill-acquisition ability always and only increasing the domain size (and difficulty).

For instance, in ARC, there is a defined set of known "core knowledge" priors (e.g. shape manipulation, counting, etc.), from which tasks are constructed. The puzzles seem relatively simple, but the enormous flexibility of conceptual instantiations and recombination makes ARC hard to generalize over [Chollet, 2019, Ellis et al., 2020]. Efficiently solving this sub-challenge of intelligence allows for concentrating development efforts on generalization while preventing runaway complexity [Chollet et al., 2025].

By narrowing the domain, models can incorporate strong assumptions or exhaustive knowledge about that domain, yielding deeper generalization within the scope (at the expense of versatility). Here, lies a strong connection with the *no free lunch theorem* we discuss more extensively in the Appendix H.4.

Takeaway: A *targeted model scope*, with sufficient coverage of relevant key primitives yet focused capabilities, yields a broad solution space while still being feasible and tractable.

3.2 (Human) Capability Encoding

(Human) Capability Encoding focuses on how human competence can be injected into a system (e.g., via curated datasets, architectural biases, or manual hyperparameter tuning). It addresses what competences are pre-loaded into a model before training. Symbolic frameworks are predetermined to incorporate human knowledge, but usually in a very rigid way, killing adaptability. For more details on generalization in symbolic systems, see appendix G. An illustrative example of how little generalization performance might be achieved despite extensive human knowledge encoding pose the top solutions to the initial ARC challenge in 2020 [icecuber, 2020, de Miquel, 2020, Larchenko, 2020].

Consequently, it is crucial to carefully focus on *abstract processes* (3.4), instead of hard-coding low-level solutions, when encoding human capabilities into a model. The idea is that the expensive extraction of high-level concepts from low-level data (3.5, 3.4) can partially be avoided making the overall systems significantly more efficient. Especially when it comes to absolute truths, such as moral (alignment), we do not want to learn from scratch, nor do we want statistical modeling. Ellis et al. [2020, p 18] emphasizes that "rich systems of built-in knowledge" radically accelerate learning – a stance aligning with the principle that *broad* competence arises from fundamental, composable operators. For ARC-AGI-1 Xu et al. [2023a] showcases how rigid function definitions can be generalized by defining *process-level* abstractions (e.g. "move(object, vector)"), resulting in reusability across countless tasks as well as a significant reduction in search space.

Takeaway: Injecting *abstract human expertise* (concept-level rather than solution-level) boosts data efficiency and encourages flexible reuse.

3.3 Meaningful Representations

Meaningful Representations are the *internal model of reality*; they define the complexity and structure of the system's "world model", whether continuous embeddings or discrete symbols. They can be engineered (e.g., ontological graphs) or emerge purely data-driven (e.g., embedding spaces). Representation spaces influence how the environment is perceived and processed, ultimately shaping what abstractions and inferences are possible. While subject to capability encoding and abstractions, they are a distinct aspect/pillar of a system.

Representational design profoundly shapes a system's ability to generalize. While neural embeddings capture latent structure, they can be overly broad for specialized tasks like ARC (3.1) [Garcez and Lamb, 2023, Skean et al., 2024]. On the other hand, graph- or object-centric representations can simplify the model's action space. For example, Xu et al. [2023a] demonstrate how a simple shift from pixel based representations to object based representations can reduce search complexity by factor 10 while simultaneously increasing model interpretability. In contrast to abstraction capabilities (3.4), which are more processing-focused, meaningful representation spaces reflect the model's perspectives on the world (i.e., world model) [Huh et al., 2024, Barbadillo, 2024].

Takeaway: Accurately aligning representations with the *natural granularity* of the domain maintains computational efficiency while setting a meaningful scope for the model.

3.4 Abstractions and Hierarchies

Abstractions and hierarchies define the ability to transform raw input into progressively more conceptual representations by discarding irrelevant details. For example, a convolutional neural network, hierarchically abstracts raw pixels into high-level semantic features. This allows to process initially different input as similar at a higher level, which is central for generalization. The system recognizes structurally similar scenarios and might transfer learned skills more readily (3.5). Latest in with AlexNet, the significance of this approach became apparent as it was able to more robustly generalize over their respective image classification tasks [Alom et al., 2018]. This flexibility is a distinct phenomenon from merely encoding a competence (3.2) or having a fitting internal representation (3.3). Layered abstractions are foundational to both human cognition and deep-network architectures [LeCun et al., 1989, Riesenhuber and Poggio, 1999, Grill-Spector and Malach, 2004, Krizhevsky et al., 2017]. In the ARC-AGI-1 context, moving from pixel-level to object- or pattern-level operations delivers major efficiency improvements [Xu et al., 2023a,b]. Each abstracted layer or module discards noisy details, accentuating shared structures across tasks while bolstering interpretability.

Takeaway: *Hierarchical design* combines low-level perception and high-level logic, enabling compositional reasoning and meaningful explanations/representations.

3.5 Knowledge Acquisition, Transfer, and Combination

No matter how thorough the initial capability encoding, novel situations inevitably appear. Thus, an intelligent system must *learn* fresh concepts during training, convert them into something useful, and *recombine* them spontaneously at inference time [Chollet, 2019]. It is debatable whether *knowledge acquisition* (how to get it) and *knowledge transfer* (how to use it) shall be considered two distinct pillars. Nevertheless, classical machine learning, primarily focuses on this pillar and has achieved major success so far.

Knowledge acquisition from data is fundamental for a generalizing system. A prerequisite are usually comprehensive training algorithms and fitting model architectures with sufficient capacity. Illustrative examples of the usefulness of knowledge acquisition is Transfer learning; for instance, a neural network pre-trained on a large dataset (like ImageNet or Wikipedia text) can be fine-tuned on a smaller target task, leveraging learned features or language understanding generalizing with less data [Weiss et al., 2016]. Also, multi-task learning utilizing shared representations (3.3) was shown to speed up the learning process [Zhang and Yang, 2018]. Multimodal LLMs follow a similar paradigm and are demonstrably applicable to many tasks/domains [Wang et al., 2024b]. Likewise, the field of meta-learning heavily relies on extracting useful information from the given data to generalize to new situations [Vettoruzzo et al., 2024].

The difference between knowledge acquisition and transfer is often fluid. However, **Knowledge transfer** from one domain to another is more involved; it requires systems to refine existing knowledge into concepts which are also useful on unseen tasks. The dependence on abstractions (3.4) is apparent, as non-abstracted knowledge is usually too situation-specific to be transferable. Knowledge transfer and recombination might happen during training as well as at inference time. For example, DreamCoder's "sleep-wake" cycle continuously refines a library of existing abstractions (dynamic concept synthesis) to transfer abstracted knowledge onto novel situations [Ellis et al., 2020]. Bober-Irizar and Banerjee [2024] demonstrates how DreamCoder can be adapted for successfully handling diverse ARC puzzles based on different underlying concepts. These approaches are primarily focused on transfer during training, not inference.

A good example for knowledge recombination during inference (as re-training/fine-tuning is too expensive) are LLM-based approaches. On ARC-AGI-1, test-time fine-tuning (Test-Time Fine-Tuning (TTFT)) has proven an essential tool for high-performing LLM-based models [Akyürek et al., 2024, Chollet et al., 2025]. The highest performing ARC-AGI-1 solution so far relies heavily on TTFT [ARC Prize, 2025a].

Takeaway: Flexible generalization arises from *continual concept formation* plus *dynamic adaptation* at test time.

3.6 Multi-Component Synergy Effects

Although some concepts are more critical, other "side problems" like uncertainty handling, and capability encoding are equally significant for broad-scope generalization [Bhuyan et al., 2024]. Many advanced methods overlook *at least one dimension* (e.g. using trivial transformations or underpowered representations), losing potential flexibility [Franzen et al., 2024, Berman, 2024]. In contrast, a systematic approach addresses each sub-component fostering powerful synergy effects between them [Garcez and Lamb, 2023].

A classical example is AlphaZero that uses a combination of Monte Carlo Tree Search (MCTS), a neural network, and a DSL to achieve superhuman performance in chess, shogi, and Go [Silver et al., 2017]. It achieves this by synergizing pillars: it uses a deep neural network to represent game states (3.3) and learn policies/value functions via self-play (3.5), but it also integrates an MCTS planner (3.4) and a capability encoding (3.2) in form of hard-coded rules of the game. The result is a system that acquires superhuman skill over multiple board games without extensively prepared training data, human expert data or handcrafted evaluations [Silver et al., 2017, McGrath et al., 2022].

On ARC, Bober-Irizar and Banerjee [2024] utilize a DreamCoder-inspired neuro-symbolic approach that significantly outperforms naive DSL search by leveraging richer learned representations (3.3) and heuristics (3.5). It integrates a symbolic program synthesis over a ARC-specific (3.1) domain specific language (3.2). Over-time it recognizes via a neural recognition model (3.3), usefull terms it abstracts (3.4) and adds to its knowledge base (3.5) [Bober-Irizar and Banerjee, 2024, Ellis et al., 2020].

Takeaway: By *holistically* optimizing each component in the system, one transcends individual contributions and achieves system-wide synergy, enabling more capable and efficient generalization.

3.7 Concluding Remarks on the Pillars

Although these pillars are inherently interrelated, each highlights a distinct mechanism supporting or constraining a system's competence to generalize over novel tasks. Collectively, the six pillars constitute the blueprint for efficient generalization. When each is addressed deliberately and woven together cohesively, the resulting system achieves far more than either paradigm alone. We posit, therefore, that *fully engaging these pillars is indispensable* for the next leap in data-efficient, developer-aware, potentially interpretable, and efficient general AI.

4 Conclusion

In this position paper, we have argued that *efficient generalization requires the deliberate integration of six fundamental pillars* (see section 3). Our analysis of historical and contemporary AI approaches indicates that the pillars are not merely optional enhancements but *crucial components* for achieving skill-acquisition efficiency. Most promising approaches to generalization already implicitly optimize for these pillars to varying degrees. By making this framework *explicit*, we provide researchers with a conceptual design principle for developing more efficient generalizing systems. On this basis, we propose three priority research directions to advance efficient generalization:

- **Holistic Generalization Benchmarking**: More comprehensive test suites to systematically measure how effective systems implement all six pillars and convert their capacities into synergistic skill-acquisition. ARC-AGI-style benchmarks go a first step, but cannot control (yet) for crucial confounding factors like data augmentation (undermining low-data constraints), training effort (e.g., compute, time, grokking, etc.) or developer competence leakage (inflating system-centric capabilities). Besides operating conditions (cost, time, compute), there is no analysis of *how* models arrive at a conclusion/output. Additionally, real-world-focused datasets will be necessary to demonstrate practical applicability and foster adoption beyond academic contexts.
- **Pillar-Aware Architecture Design**: Create modular architectures with explicit interfaces between components representing different pillars/concepts, allowing researchers to systematically study their interactions. If we want to measure the quality of our pillars, we need to concretely distinguish and collect metrics on them.
- **Harvesting Synergy Effects**: Develop methods to analyze and track the dynamic sharing of information across pillars. We might not be able to understand the complex interactions within the human brain, but for artificial systems we have a chance at understanding how their pillars interrelate and the synergy effects emerge, enabling us to leverage synergies more effectively during both training and inference.

We also found that the pillars must be developed in concert, as their *interplay and synergy effects* are what enable systems to generalize efficiently across diverse tasks with minimal data and developer engineering. While individual pillars have been researched in isolation, systems optimizing only for one or two dimensions inevitably fall short of robust generalization. This explains why purely neural or purely symbolic approaches – and even many current neuro-symbolic systems – struggle with efficient skill acquisition. At least for neuro-symbolic approaches, it has already been acknowledged that *proper* modular integration is crucial yet technically daunting [Garcez and Lamb, 2023, Chollet et al., 2025]. Similar to neuro-symbolic AI, where the individual components are not novel in isolation (neural networks, symbolic approaches), our pillars are not fundamentally new either. However, acknowledging the significance of their interconnection and the resulting effects offers a fresh lens on the **essence of generalization**. This also connects to the modular-yet-highly-integrated functional organization of the human brain, where specialized modules work in concert to create flexible cognition [Kahneman, 2011]. To our knowledge, no prior work has systematically mapped these six pillars or argued for their collective necessity in enabling true skill-acquisition efficiency. This six-pillar approach offers a systematic way to evaluate and enhance existing architectures while guiding the development of novel ones.

We urge the research community to shift focus from monolithic approaches or partial implementations toward comprehensive architectures that deliberately integrate the pillars. While developing such systems presents significant engineering challenges, we believe this path offers the most promising route to achieving truly efficient general intelligence – systems that, in the long run, can flexibly adapt to novel situations with minimal resources while potentially remaining transparent, interpretable, and reliable.

References

Stavros P. Adam, Stamatios-Aggelos N. Alexandropoulos, Panos M. Pardalos, and Michael N. Vrahatis. No Free Lunch Theorem: A Review. In Ioannis C. Demetriou and Panos M. Pardalos, editors, *Approximation and Optimization : Algorithms, Complexity and Applications*, pages 57–82. Springer International Publishing, Cham, 2019. ISBN 978-3-030-12767-1. doi: 10.1007/978-3-030-12767-1_5. URL https://doi.org/10.1007/978-3-030-12767-1_5.

Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. The Surprising Effectiveness of Test-Time Training for Abstract Reasoning, November 2024. URL http://arxiv.org/abs/2411.07279. arXiv:2411.07279 [cs].

Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esesn, Abdul A. S. Awwal, and Vijayan K. Asari. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches, September 2018. URL http://arxiv.org/abs/1803.01164. arXiv:1803.01164 [cs].

Inc. ARC Prize. ARC-AGI-1 2024 Public and Private Leaderboard, 2024. URL https://arcprize.org/leaderboard#arc-agi-pub.

Inc. ARC Prize. OpenAI o3 Breakthrough High Score on ARC-AGI-Pub, 2025a. URL https://arcprize.org/blog/oai-o3-pub-breakthrough.

Inc. ARC Prize. Analyzing o3 and o4-mini with ARC-AGI, 2025b. URL https://arcprize.org/blog/analyzing-o3-with-arc-agi.

Inc. ARC Prize. ARC-AGI-2 + ARC Prize 2025 is Live!, 2025c. URL https://arcprize.org/blog/announcing-arc-agi-2-and-arc-prize-2025.

Inc. ARC Prize. ARC-AGI Leaderboard, 2025d. URL https://arcprize.org/leaderboard.

Inc. ARC Prize. An Analysis of DeepSeek's R1-Zero and R1, 2025e. URL https://arcprize.org/blog/r1-zero-r1-results-analysis.

Inc. ARC Prize. ARC Prize Side Quest: SnakeBench, 2025f. URL https://arcprize.org/blog/snakebench.

Marthe Ballon, Andres Algaba, and Vincent Ginis. The Relationship Between Reasoning and Performance in Large Language Models – o3 (mini) Thinks Harder, Not Longer, February 2025. URL https://arxiv.org/abs/2502.15631v1.

Guillermo Barbadillo. Solution Summary, 2024. URL https://ironbar.github.io/arc24/05_Solution_Summary/.

Jeremy Berman. How I came in first on ARC-AGI-Pub using Sonnet 3.5 with Evolutionary Test-time Compute, 2024. URL https://jeremyberman.substack.com/p/how-i-got-a-record-536-on-arc-agi.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i16.29720. URL https://ojs.aaai.org/index.php/AAAI/article/view/29720. Number: 16.

Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and T. P. Singh. Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications*, 36(21):12809–12844, July 2024. ISSN 1433-3058. doi: 10.1007/s00521-024-09960-z. URL https://doi.org/10.1007/s00521-024-09960-z.

Mikel Bober-Irizar and Soumya Banerjee. Neural networks for abstraction and reasoning. *Scientific Reports*, 14(1):27823, November 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-73582-7. URL https://www.nature.com/articles/s41598-024-73582-7. Publisher: Nature Publishing Group.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4, April 2023. URL http://arxiv.org/abs/2303.12712. arXiv:2303.12712 [cs].

Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. ARC Prize 2024: Technical Report, January 2025. URL http://arxiv.org/abs/2412.04604. arXiv:2412.04604 [cs].

François Chollet. On the Measure of Intelligence, November 2019. URL http://arxiv.org/abs/1911.01547. arXiv:1911.01547 [cs].

Artur S. d'Avila Garcez and Luis C. Lamb. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, November 2023. ISSN 1573-7462. doi: 10.1007/s10462-023-10448-w.

Alejandro de Miquel. 2020 kaggle arc-agi-1 challange: 2nd place solution, 2020. URL https://www.kaggle.com/competitions/abstraction-and-reasoning-challenge/discussion/154391.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025. URL http://arxiv.org/abs/2501.12948. arXiv:2501.12948 [cs].

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A Survey on In-context Learning, June 2023. URL http://arxiv.org/abs/2301.00234. arXiv:2301.00234 [cs].

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and Fate: Limits of Transformers on Compositionality, June 2023. URL http://arxiv.org/abs/2305.18654. arXiv:2305.18654 [cs].

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: Growing generalizable,

interpretable knowledge with wake-sleep Bayesian program learning, June 2020. URL http://arxiv.org/abs/2006.08381. arXiv:2006.08381 [cs].

Tom Everitt and Marcus Hutter. Universal Artificial Intelligence. In Hussein A. Abbass, Jason Scholz, and Darryn J. Reid, editors, *Foundations of Trusted Autonomy*, pages 15–46. Springer International Publishing, Cham, 2018. ISBN 978-3-319-64816-3. doi: 10.1007/978-3-319-64816-3_2. URL https://doi.org/10.1007/978-3-319-64816-3_2.

Jonathan Feldstein, Paulius Dilkas, Vaishak Belle, and Efthymia Tsamoura. Mapping the Neuro-Symbolic AI Landscape by Architectures: A Handbook on Augmenting Deep Learning Through Symbolic Reasoning, October 2024. URL http://arxiv.org/abs/2410.22077. arXiv:2410.22077 [cs].

Daniel Franzen, Jan Disselhoff, and David Hartmann. The LLM ARChitect: Solving ARC-AGI Is A Matter of Perspective, 2024. URL https://github.com/da-fr/arc-prize-2024/blob/main/the_architects.pdf.

Artur d'Avila Garcez and Luís C. Lamb. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, November 2023. ISSN 1573-7462. doi: 10.1007/s10462-023-10448-w. URL https://doi.org/10.1007/s10462-023-10448-w.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are We Done with MMLU?, June 2024. URL https://arxiv.org/abs/2406.04127v3.

Micah Goldblum, Marc Anton Finzi, Keefer Rowan, and Andrew Gordon Wilson. Position: The No Free Lunch Theorem, Kolmogorov Complexity, and the Role of Inductive Biases in Machine Learning. June 2024. URL https://openreview.net/forum?id=EaJ7nqJ2Fa.

Ryan Greenblatt. Getting 50% (SoTA) on ARC-AGI with GPT-4o, 2024a. URL https://redwoodresearch.substack.com/p/getting-50-sota-on-arc-agi-with-gpt.

Ryan Greenblatt. Getting 50% (SoTA) on ARC-AGI with GPT-4o, June 2024b. URL https://redwoodresearch.substack.com/p/getting-50-sota-on-arc-agi-with-gpt.

Kalanit Grill-Spector and Rafael Malach. THE HUMAN VISUAL CORTEX. *Annual Review of Neuroscience*, 27(Volume 27, 2004):649–677, July 2004. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.27.070203.144220. URL https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.27.070203.144220. Publisher: Annual Reviews.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. *Semantic Web*, Preprint (Preprint):1–42, January 2022. ISSN 1570-0844. doi: 10.3233/SW-223228. URL https://content.iospress.com/articles/semantic-web/sw223228. Publisher: IOS Press.

Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation*, 16 (1):45–74, January 2024. ISSN 1866-9964. doi: 10.1007/s12559-023-10179-8. URL https://doi.org/10.1007/s12559-023-10179-8.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding, January 2021. URL http://arxiv.org/abs/2009.03300. arXiv:2009.03300 [cs].

José Hernández-Orallo. Twenty Years Beyond the Turing Test: Moving Beyond the Human Judges Too. *Minds and Machines*, 30(4):533–562, December 2020. ISSN 1572-8641. doi: 10.1007/s11023-020-09549-0. URL https://doi.org/10.1007/s11023-020-09549-0.

Pascal Hitzler, Aaron Eberhart, Monireh Ebrahimi, Md Kamruzzaman Sarker, and Lu Zhou. Neuro-symbolic approaches in artificial intelligence. *National Science Review*, 9(6):nwac035, June 2022. ISSN 2095-5138. doi: 10.1093/nsr/nwac035. URL https://doi.org/10.1093/nsr/nwac035.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The Platonic Representation Hypothesis, July 2024. URL http://arxiv.org/abs/2405.07987. arXiv:2405.07987 [cs].

icecuber. 2020 kaggle arc-agi-1 challange: 1st place solution + code and official documentation, 2020. URL https://www.kaggle.com/competitions/abstraction-and-reasoning-challenge/discussion/154597.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and Applications of Large Language Models, July 2023. URL http://arxiv.org/abs/2307.10169. arXiv:2307.10169 [cs].

Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

J. K. Kastner and S. J. Hong. A review of expert systems. *European Journal of Operational Research*, 18(3):285–292, December 1984. ISSN 0377-2217. doi: 10.1016/0377-2217(84)90150-4. URL https://www.sciencedirect.com/science/article/pii/0377221784901504.

M. Keber, I. Grubišić, A. Barešić, and A. Jović. A Review on Neuro-symbolic AI Improvements to Natural Language Processing. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 66–72, May 2024. doi: 10.1109/MIPRO60963.2024.10569741. URL https://ieeexplore.ieee.org/abstract/document/10569741. ISSN: 2623-8764.

Eoin M Kenny, Courtney Ford, Molly Quinn, and Mark T Keane. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 294:103459, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. ISSN 0001-0782, 1557-7317. doi: 10.1145/3065386. URL https://dl.acm.org/doi/10.1145/3065386.

Ilia Larchenko. 2020 Kaggle ARC-AGI-1 Challange: My part of the 3rd place solution, 2020. URL https://www.kaggle.com/competitions/abstraction-and-reasoning-challenge/discussion/154409.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1 (4):541–551, December 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL https://ieeexplore.ieee.org/abstract/document/6795724. Conference Name: Neural Computation.

Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci. When are post-hoc conceptual explanations identifiable? In *Uncertainty in Artificial Intelligence*, pages 1207–1218. PMLR, 2023.

Shane Legg and Marcus Hutter. Universal Intelligence: A Definition of Machine Intelligence, December 2007. URL http://arxiv.org/abs/0712.3329. arXiv:0712.3329 [cs].

Martha Lewis and Melanie Mitchell. Using Counterfactual Tasks to Evaluate the Generality of Analogical Reasoning in Large Language Models, February 2024. URL http://arxiv.org/abs/2402.08955. arXiv:2402.08955.

Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal Foundation Models: From Specialists to General-Purpose Assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, May 2024. ISSN 1572-2740, 1572-2759. doi: 10.1561/0600000110. URL https://www.nowpublishers.com/article/Details/CGV-110. Publisher: Now Publishers, Inc.

Sanae Lotfi, Marc Finzi, Yilun Kuang, Tim G. J. Rudner, Micah Goldblum, and Andrew Gordon Wilson. Non-Vacuous Generalization Bounds for Large Language Models, July 2024. URL http://arxiv.org/abs/2312.17173. arXiv:2312.17173.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540, June 2024. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2024.01.011. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(24)00027-5. Publisher: Elsevier.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in AlphaZero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, November 2022. doi: 10.1073/pnas.2206625119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2206625119. Publisher: Proceedings of the National Academy of Sciences.

José Mira Mira. Symbols versus connections: 50 years of artificial intelligence. *Neurocomputing*, 71(4):671–680, January 2008. ISSN 0925-2312. doi: 10.1016/j.neucom.2007.06.009. URL https://www.sciencedirect.com/science/article/pii/S0925231207003451.

Arseny Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. The ConceptARC Benchmark: Evaluating Understanding and Generalization in the ARC Domain, May 2023. URL http://arxiv.org/abs/2305.07141. arXiv:2305.07141 [cs].

Allen Newell. YOU CAN'T PLAY 20 QUESTIONS WITH NATURE AND WIN: PROJECTIVE COMMENTS ON THE PAPERS OF THIS SYMPOSIUM. In *Visual Information Processing*, pages 283–308. Elsevier, 1973. ISBN 978-0-12-170150-5. doi: 10.1016/B978-0-12-170150-5.50012-3. URL https://linkinghub.elsevier.com/retrieve/pii/B9780121701505500123.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark, November 2023. URL http://arxiv.org/abs/2311.12022. arXiv:2311.12022 [cs].

Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, November 1999. ISSN 1546-1726. doi: 10.1038/14819. URL https://www.nature.com/articles/nn1199_1019. Publisher: Nature Publishing Group.

Yao Rong, Tobias Leemann, Thai-Trang Nguyen, Lisa Fiedler, Peizhu Qian, Vaibhav Unhelkar, Tina Seidel, Gjergji Kasneci, and Enkelejda Kasneci. Towards human-centered explainable ai: A survey of user studies for model explanations. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to Train Data-Efficient LLMs, February 2024. URL http://arxiv.org/abs/2402.09668. arXiv:2402.09668 [cs].

David Schlangen. Targeting the Benchmark: On Methodology in Current Natural Language Processing Research. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 670–674, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.85. URL https://aclanthology.org/2021.acl-short.85.

Dale Schuurmans, Hanjun Dai, and Francesco Zanini. Autoregressive Large Language Models are Computationally Universal, October 2024. URL http://arxiv.org/abs/2410.03170. arXiv:2410.03170.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, December 2017. URL https://arxiv.org/abs/1712.01815v1.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. Does Representation Matter? Exploring Intermediate Layers in Large Language Models, December 2024. URL http://arxiv.org/abs/2412.09563. arXiv:2412.09563 [cs].

Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in neural information processing systems*, 34: 9391–9404, 2021.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour,

Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models, June 2023. URL http://arxiv.org/abs/2206.04615. arXiv:2206.04615 [cs, stat].

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. SciEval: A Multi-Level Large Language Model Evaluation Benchmark for Scientific Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19053–19061, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i17.29872. URL https://ojs.aaai.org/index.php/AAAI/article/view/29872. Number: 17.

Ilya Sutskever. Sequence to sequence learning with neural networks: what a decade. NeurIPS 2024, 2024. URL https://www.youtube.com/watch?v=WQQdd6qGxNs.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, January 2024. ISSN 1476-4687. doi: 10.1038/s41586-023-06747-5. URL https://www.nature.com/articles/s41586-023-06747-5. Publisher: Nature Publishing Group.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can Large Language Models Really Improve by Self-critiquing Their Own Plans?, October 2023. URL http://arxiv.org/abs/2310.08118. arXiv:2310.08118 [cs].

Anna Vettoruzzo, Mohamed-Rafik Bouguelia, Joaquin Vanschoren, Thorsteinn Rögnvaldsson, and KC Santosh. Advances and Challenges in Meta-Learning: A Technical Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4763–4779, July 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.3357847. URL https://ieeexplore.ieee.org/abstract/document/10413635.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization, May 2024a. URL http://arxiv.org/abs/2405.15071. arXiv:2405.15071 [cs].

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the Reasoning Abilities of Multimodal Large Language Models (MLLMs): A Comprehensive Survey on Emerging Trends in Multimodal Reasoning, January 2024b. URL http://arxiv.org/abs/2401.06805. arXiv:2401.06805 [cs].

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent Analogical Reasoning in Large Language Models, August 2023. URL http://arxiv.org/abs/2212.09196. arXiv:2212.09196.

Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, May 2016. ISSN 2196-1115. doi: 10.1186/s40537-016-0043-6. URL https://doi.org/10.1186/s40537-016-0043-6.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Search-in-the-Chain: Interactively Enhancing Large Language Models with Search for Knowledge-intensive Tasks, February 2024. URL http://arxiv.org/abs/2304.14732. arXiv:2304.14732 [cs].

Yudong Xu, Elias B. Khalil, and Scott Sanner. Graphs, Constraints, and Search for the Abstraction and Reasoning Corpus. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4): 4115–4122, June 2023a. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i4.25527. URL https://ojs.aaai.org/index.php/AAAI/article/view/25527.

Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B. Khalil. LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations, May 2023b. URL http://arxiv.org/abs/2305.18354. arXiv:2305.18354 [cs].

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, May 2023. URL http://arxiv.org/abs/2305.10601. arXiv:2305.10601 [cs].

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. Towards Better Chain-of-Thought Prompting Strategies: A Survey, October 2023. URL http://arxiv.org/abs/2310.04959. arXiv:2310.04959 [cs].

Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1): 30–43, January 2018. ISSN 2095-5138. doi: 10.1093/nsr/nwx105. URL https://doi.org/10.1093/nsr/nwx105.

A Behaviorism vs. Internalism

A longstanding philosophical debate pertains to whether only *external* behavior matters (behaviorism) or whether the *internal* mechanisms of thought carry essential explanatory value (internalism). In contemporary machine learning, this tension appears as "functionality vs. interpretability" or "black-box vs. transparent systems." High-performing but opaque models – like many Large Language Models – demonstrate that achieving sophisticated outputs does not necessarily illuminate the process by which the model reasons [Hernández-Orallo, 2020, Schlangen, 2021].

As these systems are deployed in sensitive or high-stakes environments, interpretability and control become paramount [Hassija et al., 2024]. Post-hoc explanations often provide only a partial window into massive parameter spaces, leaving significant uncertainties about *why* a particular decision was reached [Kenny et al., 2021, Slack et al., 2021, Leemann et al., 2023, Rong et al., 2023]. By contrast, **inherent model transparency** – via symbolic modules, meaningful structured interfaces, or modular architectures – can yield more reliable comprehension of internal processes, facilitate debugging, and bolster trustworthiness. Consequently, we argue that *internalist* considerations should shape the development of any model that aspires to broader, more systematic intelligence.

B Developer-Aware Generalization

Even when a model attains notable performance on a suite of tasks, it is crucial to distinguish between *intrinsic generalization* and *engineered solutions*.

Many recent successes hinge on massive data curation, architectural tuning, or manual injection of priors – leading to impressive system-centric results, but not necessarily reflecting a model's capacity to *autonomously* learn how to solve unseen tasks. A "**developer-aware**"[5] perspective on skill acquisition controls for these extra-human interventions [Chollet, 2019]. In contrast, a *developer-centric* measure includes both the developer's competence as well as the model's competence.

Without this distinction, higher benchmark scores might be misinterpreted as an increase in the system's general intelligence, while instead, only the development process was optimized (e.g., more optimal hyperparameters do not make the model design more fitting, but the performance improves). This even applies to generality-focused benchmarks such as the Abstraction and Reasoning Corpus.

C LLM's worth for understanding intelligence

The LLMs mentioned in Section 2.1 are even less suitable as an *academic research framework* for understanding the *mechanisms* behind generalization, which are as unresolved as they are crucial. Large data combined with sufficient computing resources can brute force solutions, but they do not illuminate the core processes underlying abstract reasoning. For those interested in deeper interpretability, explainability, or developer-aware skill acquisition, *neuro-symbolic integration* might be indispensable.

D ARC-AGI limitations and alternative benchmarks

Relying heavily on the ARC benchmark(s) often appears limiting; therefore, we will shortly discuss prominent limitations, how other popular benchmarks compare, and why we still conclude that ARC is the best choice.

D.1 Limitation: Visual Data

A known caveat of ARC is that it "only" tests for **geometric problem solving via image data**. This potentially puts heavily text-oriented models at a disadvantage. Transformers are indeed very sensitive to the input/output prompting of ARC-style tasks [Greenblatt, 2024b]. Fortunately, the trend of foundation models goes towards more modality agnostic models (i.e., multi-modal models) [Li et al., 2024, Wang et al., 2024b]. Especially given the *the platonic representation hypothesis* by Huh

[5] As defined by Chollet [2019] p. 10: "Developer-aware generalization: this is the ability of a system [...], to handle situations that neither the system nor the developer of the system has encountered before." Further, "Note that 'developer-aware generalization' accounts for any prior knowledge that the developer of the system has injected into it."

et al. [2024], the concrete modality of data will lose its relevance for highly capable/universal models, making the visual nature of ARC tasks irrelevant in the future.

D.2 Limitation: Too artificial and not real-world enough

ARC is often perceived as overly structured or "toy-like." We argue this is not a limitation but a hallmark of isolating the **gist of a problem**. ARC specifically tests for generalization while minimizing all other factors. This does not mean that crucial properties are missing; for example, there exist multiple ARC tasks specifically focused on ambiguity and noise removal.

In our opinion, "real-world" data does not have any additional properties relevant for generalization. Similarly, Adam et al. [2019] came to the conclusion that the complexity of real-world tasks is highly overestimated and the prevalent meta-structure in real data is heavily underestimated.

Furthermore, unspecified *noise may obscure* whether a model genuinely generalizes or simply leverages massive data, shortcuts, or pattern memorization. ARC helps evaluate generalization in a controlled setting, offering valuable insights before tackling noisy real-world environments. Given the early research stage of skill-acquisition-focused approaches, an overly unstructured setting is *counterproductive* for progress.

Furthermore, we believe that researchers unfamiliar with ARC tend to significantly **overestimate the structure** in the actual ARC challenge. Between tasks, there is no structural similarity, resulting in a lack of invariants between tasks (except the underlying geometric rules). If there were easily exploitable structural patterns across ARC tasks, traditional neural networks and feature extractors would work significantly better on ARC, but they do not. Consequently, on ARC-AGI-2, even LRMs have major issues improving beyond a single-digit performance count [ARC Prize, 2025c].

While practical deployment requires **large-scale integration**, lessons from ARC can inform real-world contexts where robust, data-efficient learning is critical. Consequently, the insights gained on ARC are also valuable for the real world. Implementing them in real-world systems is primarily an engineering challenge (and not the scope of this position paper).

D.3 Alternatives

ARC is not the only dataset targeting higher-level reasoning, but it is the only one that explicitly stresses *skill-acquisition efficiency*. For completeness' sake, we want to give some details on why other, popular "reasoning" benchmarks are less suitable regarding efficient skill acquisition. Table 1 provides an overview of the most prominent alternatives (selection is not exhaustive and potentially subjective).

Benchmark	Core Skill Tested	Efficiency Gap
ARC	Geometric problem solving, abstract reasoning	Explicitly designed to measure skill-acquisition efficiency. On ARC-AGI-2, Humans: 98%, LLMs : <4% [ARC Prize, 2025d]
BIG-Bench	Mixed reasoning, code, riddles	Many-shot CoT sampling lets models brute-force via scale [Srivastava et al., 2023]
MMLU	Broad domain knowledge	Scores dominated by pre-training, not in-task learning [Hendrycks et al., 2021]
SciEval	Dynamic STEM QA	Unlimited query passes hide inference compute cost [Sun et al., 2024]
GPQA	Expert closed-book retrieval	Measures "knows," not "learns"; retrieval > abstraction [Rein et al., 2023]

Table 1: Popular reasoning benchmarks versus *skill-acquisition efficiency*. Each still rewards brute-force scale or developer engineering, obscuring the cost–competence ratio that ARC measures.

We shortly describe each benchmark and where we think they are lacking:

- **BIG-Bench** (Beyond the Imitation Game Benchmark), is a collaborative benchmark with 200+ tasks such as code debugging, riddles, and obscure language translation. It is supposed to test robustness to ambiguity and compositional reasoning through diverse challenges requiring logic, creativity, and cross-domain knowledge transfer [Srivastava et al., 2023].

It includes many complex reasoning tasks and even "challenge" tasks beyond current AI, some of which require reasoning steps or creative generalization, testing *flexibility* to an extent. However, models are evaluated in a zero- or few-shot prompt setting, and many tasks still correlate with knowledge or patterns seen in training. Indeed, tasks with a large knowledge component show gradual improvement as model size increases (suggesting *pattern-learning*), whereas truly novel multi-step reasoning tasks remain unsolved until models hit a scale "breakthrough" point. In short, BIG-Bench does probe generalization, but it does not uniformly enforce *learning new skills* – some tasks can be partially solved by pattern matching or sheer pre-trained knowledge [Srivastava et al., 2023].

- **MMLU** (Massive Multitask Language Understanding), covers 57 subjects across STEM, humanities, and social sciences to assess broad knowledge application. It is supposed to measure how models generalize across disciplines rather than excelling at narrow tasks [Hendrycks et al., 2021]. MMLU is knowledge-centric and less focused on new skill learning. It primarily measures factual knowledge and some reasoning acquired from large-scale training. There is little focus on adapting to *novel* tasks as questions resemble those in textbooks or exams. Strong model performance often comes from more training data or model parameters, not *on-the-fly abstraction*. Additionally, MMLU seems to suffer from dataset quality and transparency issues [Gema et al., 2024].
- **SciEval** (Scientific Evaluation Benchmark), uses dynamically generated questions in physics, chemistry, and biology to prevent memorization. Models shall demonstrate genuine scientific reasoning, with GPT-4's accuracy dropping from 65% to 26% when tested on novel dynamic data [Sun et al., 2024]. It is valuable for assessing in-depth reasoning in science, requiring models to apply scientific knowledge and logic rather than just memorize answers. The inclusion of dynamically generated problems means models face some novel content, revealing whether they can reason beyond rote memory (GPT-4 still has "substantial room for improvement" on these dynamic questions [Sun et al., 2024]). However, SciEval is limited to the scientific domain – it measures generalization *within* science (e.g., applying known principles in new ways) more than general skill acquisition across arbitrary tasks. Models are not learning entirely new kinds of tasks; they are answering science questions (albeit challenging ones) using prior scientific knowledge. This is a test of *expertise* and *reasoning depth*, but not *learning to learn* broadly outside the science context [Sun et al., 2024].
- **GPQA** (Graduate-Level Google-Proof QA), features 448 expert-level multiple-choice questions in biology, physics, and chemistry where internet access provides minimal human performance gains (34% vs 25% baseline). This benchmark tests for deep conceptual understanding rather than information retrieval capabilities [Rein et al., 2023]. It specifically targets questions that demand (in humans) complex reasoning or deep understanding, beyond simple fact recall. It highlights the gap between human experts and AI on truly *hard* questions. In terms of skill acquisition, though, GPQA remains a static question benchmark. The model is not asked to *learn* a new skill; it is challenged to apply its existing high-level knowledge in novel, intricate ways. Failing GPQA often means the AI lacks the necessary combined knowledge or reasoning chain, rather than failing to learn from small data (since no new training is given). Thus, GPQA is excellent for stress-testing an AI's reasoning within known domains, but it does not evaluate the process of *rapidly learning* an entirely new type of task or concept [Rein et al., 2023].

Most benchmarks **assume** that some aspect of human intelligence is required to perform well on the given dataset, simply because humans would require intelligence to solve it. A theoretical foundation for such assumptions is usually lacking. Historically, *chess* was also once considered a real-world intelligence benchmark as humans require a diverse repertoire of skills to solve it (tactics, reasoning, planning, means-end analysis, theory of mind, deception, etc.) [Newell, 1973]. However, it later became obvious that a naive tree-search method - lacking all signs of intelligence - is perfectly capable of solving chess. We are therefore cautious not to overestimate the value of "established" benchmarks of intelligent agents that do not have an explicit grounding regarding skill-acquisition efficiency.

To cite the ARC foundation themselves (ARC Prize [2025c]):

> *All other AI benchmarks focus on superhuman capabilities or specialized knowledge by testing 'PhD++' skills. ARC-AGI is the only benchmark that takes the opposite design choice – by focusing on tasks that are relatively easy for humans,*

yet hard, or impossible, for AI, we shine a spotlight on capability gaps that do not spontaneously emerge from "scaling up".

As Chollet [2019] substantiated ARC with an extensive theory of how to measure generalization, we find the ARC challenge to be much more valuable for progress on AGI than any currently existing *"real-world"* benchmark. Of course, we are aware that this claim is up for discussion and remains to be proven right or wrong with time.

D.4 Conclusion

ARC it is one of the few benchmarks explicitly designed to minimize statistical shortcuts and **emphasize skill-acquisition efficiency**. By removing extraneous noise and avoiding easily exploitable patterns, ARC isolates how effectively a model can acquire and transfer skills from limited experience. Pinpointing and measuring skill acquisition efficiency is central to generalization (see 1) and a key research question in itself. To the best of our knowledge, ARC is the most relevant benchmark regarding this notion of generalization, which is why it's our priority.

While ARC is not perfect, it provides an environment to observe the interplay of multiple factors (architectural choices, knowledge priors, hierarchical abstractions, etc.), providing an ideal testbed for novel generalization approaches, fostering progress.

E Dimensions for designing models

Certain system properties (e.g., transparency or interpretability) are not strictly required for a model to generalize well on some tasks. However, as discussed in Section A, one should focus beyond mere performance. To offer design guidance for more robust, reliable, and maintainable systems we provide guardrails for the developmental process in Table 2. These are suggested design principles rather than mandatory criteria.

Table 2: Key Dimensions for Designing Models with Broad Generalization

Dimension	Importance for General Intelligence	Representative Works
Skill-Acquisition Efficiency	Emphasizes how well a system adapts to new tasks without extensive retraining; penalizes overreliance on developer engineering or huge datasets.	Chollet [2019], Bober-Irizar and Banerjee [2024]
Transparency & Interpretability	Strengthens trust and debugging; post-hoc explanations are often insufficient for large black-box models. Inherent transparency is crucial for real-world reliability.	Hernández-Orallo [2020], Hassija et al. [2024]
Symbolic Reasoning	Allows compositional, logically coherent transformations. Fosters human-level abstraction and provides robust handling of discrete structures.	d'Avila Garcez and Lamb [2023], Keber et al. [2024]
Neural Representations	Harnesses powerful pattern-extraction capabilities from raw data (images, text), enabling feature discovery and capturing nuanced correlations.	Bubeck et al. [2023]
Small-Data Adaptation	Avoids brute-forcing solutions by demanding strong generalization from very few examples (as in ARC tasks), exposing true abstraction capabilities.	Moskvichev et al. [2023], Chollet et al. [2025]

F Representative Neuro-Symbolic Approaches

Table 3 summarizes representative state-of-the-art neuro-symbolic approaches that have been shown to be effective for generalization in ARC-like tasks.

G Purely Symbolic Approaches: Domain-Specific Languages and Program Synthesis

Although overshadowed by neural methods in recent years, purely symbolic or logic-based AI once dominated AI research and retains a devoted following Kastner and Hong [1984]. To provide an overview of their limitations for current generalization challenges we analyze them in context of

Table 3: Representative Neuro-Symbolic Approaches for Generalization in ARC-like Tasks

Approach	Neural Component	Symbolic Component	Key Mechanism & Insights
Bober-Irizar & Banerjee (2024) [Bober-Irizar and Banerjee, 2024]	Learned concept-formation module (e.g., CNN-like embeddings to identify object features)	DSL-based program search for transformations	Uses neural heuristics to guide symbolic search, significantly reducing the DSL's combinatorial explosion. Demonstrates notable gains on ARC tasks versus purely symbolic baselines.
DreamCoder [Ellis et al., 2020]	Neural "wake-sleep" cycle that learns common subroutines or concepts	Inductive program synthesis in a high-level language (with control-flow, recursion)	Iteratively refines a library of reusable functions – symbolic *abstractions* – guided by neural scoring. *DreamCoder* is not specifically designed for ARC but illustrates how learned domain knowledge can be symbolically encoded.
Neuro-Symbolic DSL Enhancements (various) [Hamilton et al., 2022, Hitzler et al., 2022, Garcez and Lamb, 2023, Bhuyan et al., 2024]	Neural embeddings for object detection, classification, or spatial feature extraction	Logic-based DSL or ontology enforcing compositional rules	General family of hybrid methods: neural modules handle perceptual tasks or fuzzy matches, while symbolic DSL enforces interpretability and constraint satisfaction. Shown to improve data-efficiency and interpretability on small "grid-world" or ARC-like puzzles.

ARC-AGI-1. Within the ARC domain, the most visible symbolic attempts revolve around exhaustive search in a *Domain-Specific Language* (DSL) or program-synthesis methods such as DreamCoder Ellis et al. [2020].

DSL-Based Methods. Early top-ranked solutions in the original ARC challenge relied on large, hand-crafted DSLs icecuber [2020], de Miquel [2020], Larchenko [2020]. By systematically searching over a predefined set of transformations and heuristics, these approaches found valid transformations for specific puzzles. However, these DSL-based methods achieved only modest coverage due to the combinatorial explosion of possible transformations and the diversity of ARC tasks. They also demanded extensive human engineering to hard-code each concept, undermining *developer-aware* generalization measures Bober-Irizar and Banerjee [2024].

Program Synthesis Approaches. Program-synthesis frameworks like DreamCoder Ellis et al. [2020] extend the DSL idea with higher-level constructs (e.g., control-flow operators, recursion). While this unlocks greater expressiveness, it can also inflate the search space. Adapting a fully general programming language for ARC tasks becomes cumbersome because ARC-AGI-1 is already quite challenging without further increasing the solution space Bober-Irizar and Banerjee [2024].

Symbolic Drawbacks. While symbolic approaches can offer strong interpretability (one can often track each logical step explicitly), they typically struggle to infer abstract "core concepts" from limited data without some learned inductive biases. Their purely top-down logic has trouble coping with the noisy, high-dimensional input distributions where data-driven feature extraction is crucial. Additionally, naive symbolic search tends to be fragile in the face of tasks requiring approximate or probabilistic reasoning. A ubiquitous problem of symbolic approaches - even outside of ARC - is the above-mentioned complexity explosion, making scalability to real-world settings often infeasible [Garcez and Lamb, 2023].

Conclusion Historically, purely symbolic solutions have rarely scaled well across diverse tasks and have difficulty encoding robust priors for low-data settings Kastner and Hong [1984], Ellis et al. [2020]. Conversely, the golden era of symbolic AI faded in the late 1980s, giving way to sub-symbolic (neural) approaches. Still, the ARC challenge confirms that exhaustive or highly engineered symbolic DSLs rapidly reach diminishing returns. Hence, purely symbolic approaches, while valuable for interpretability and logic, alone are still insufficient for broad or efficient generalization.

The limitations that once suffocated symbolic AI – such as brittle rule systems or exponential search complexity – can be mitigated by modern neural advances and computing power Mira [2008]. However, those hybrid, neuro-symbolic approaches go beyond what we consider purely symbolic.

H Frequently Asked Questions

H.1 General Paper Structure and Approach

Your paper reads like a survey, are you sure it's a position paper? Yes. We intentionally include numerous references to prior literature as necessary grounding for our claims. Much research about AGI is (highly) speculative. Therefore, we carefully selected many different sources to substantiate our claims. Nevertheless, we are not only stating previous developments, but also distill a structured perspective on how to possibly achieve efficient generalization, which we would like the research community to consider more intensively.

This is just a general description of desiderata. Why is it so vague? We take a meta-level perspective, aiming to shift focus toward systematically addressing six core components (see Section 3) of generalization. We aim to articulate a conceptual framework of guardrails and design guidelines relevant for achieving robust skill-acquisition and generalization. Concrete work implementing multiple pillars to varying degrees already exists. Therefore, our motivation is to guide researchers toward systematically and consciously using these components to enable synergy effects resulting in effective generalization. Our paper shall broaden the conversation from "LLMs vs. Neuro-Symbolic" to "How do we achieve efficient generalization, and what is essential to that end?". We contribute a mindset shift for the research community (research direction), not concrete experimental results/insights.

Consequently, it is up to the ML community to further conceive, test, and refine systems specifically focused on these pillars. We provide a design paradigm that must be instantiated and therefore do not provide extensive algorithmic details, which would be more appropriate for a concrete research-track paper.

Why focus on ARC when it's just a visual reasoning benchmark? We address this more extensively in Appendix D. In short, ARC is not merely a visual reasoning benchmark but a carefully designed test for generalization capability that minimizes statistical shortcuts and emphasizes skill-acquisition efficiency. The visual modality is incidental, not central – what matters is the benchmark's ability to isolate generalization from other confounding factors. With the trend toward more modality-agnostic models and the Platonic Representation Hypothesis [Huh et al., 2024], the specific modality becomes increasingly irrelevant for highly capable models anyway.

H.2 Six Pillars Framework

How do your six pillars differ from neuro-symbolic frameworks? While neuro-symbolic approaches typically focus on combining neural and symbolic components, our six-pillar framework takes a more comprehensive view. Existing frameworks tend to emphasize the neural-symbolic integration itself, whereas our pillars address multiple orthogonal aspects of generalization: from model specificity and (human) capability encoding to meaningful representation spaces and abstract hierarchies. Most of the pillars can be implemented either in a purely neuronal or purely symbolic way (even if that does not make much sense in practice).

Our framework also explicitly highlights the synergy effects between these components in terms of skill-acquisition efficiency rather than just performance or capability, prioritizing generalization from minimal data and experience.

What concrete evidence supports the necessity of all six pillars? The need for all six pillars is supported by analyzing the limitations of current approaches that excel in some areas but fail in others. For instance, purely neural approaches lack compositional reasoning (addressed by Multi-Component Synergy), while purely symbolic methods struggle with meaningful representation learning and adaptation (addressed by Knowledge Acquisition & Transfer). The success of approaches incorporating multiple pillars, such as the neuro-symbolic method by Bober-Irizar and Banerjee [2024], provides empirical evidence for their collective importance. However, to the best of our knowledge, no single system has yet fully optimized all six pillars simultaneously, which represents a key research frontier. Each pillar addresses distinct failure modes observed in existing AI systems when confronted with generalization challenges.

How do you measure "efficiency" in skill acquisition? We adopt the framework proposed by Chollet [2019], where efficiency is measured as the ratio between the competence gained and the resources required. Specifically, we consider: (1) data efficiency – how much performance is achieved from minimal examples; (2) computational efficiency – the processing resources needed during both

training and inference; (3) developer effort – controlling for human engineering in the system design; and (4) transfer capacity – how well skills learned in one context apply to novel scenarios. That the ARC benchmark (organizers) cannot control all the relevant factors (i.e., data augmentation, developer effort, training compute) is an open issue/limitation, which - however - applies to most benchmarks.

In an optimal world, an efficient system would solve novel ARC tasks with minimal resources while a less efficient system (like current LLMs) might achieve similar performance but at vastly greater computational or data costs. Chollet [2019] also proposed some formalisms how to calculate the skill-aquisition efficiency and mathematically correct for confounding factors (see Section II.2.2 of [Chollet, 2019]).

H.3 LLMs and Scaling

For AGI, why don't we just use LLMs, and make them more powerful? You already suggested extending LLMs with symbolic scaffolding to improve their lack of reasoning. You can do that, and you will probably also get something that looks like AGI at some point. But that has not much to do with skill-acquisition efficiency. Making these approaches more efficient afterwards might be very hard, so we suggest we start with efficiently generalizing systems first and steadily expand the scope.

We will make LLMs more efficient at some point. Computers were also building-sized and now are small. Maybe you can increase the efficiency of a LLM-based system by the same order of magnitude as hardware improves (see Moore's law). But maybe we cannot, as there might be more fundamental issues. We will see with time. In the meantime, our position paper proposes a parallel research direction that is inherently focused on generality by design from the start.

As a reference, until ARC-AGI-1 could be solved with near-human performance, it required Large Reasoning Models. These models are currently at a low two-digit percentage on ARC-AGI-2 [ARC Prize, 2025c,b]. It will take some time (and effort) to crack ARC-AGI-2, not even mentioning ARC-AGI-3, which is also on its way [ARC Prize, 2025c]. This being said; ARC only tackles relatively straightforward geometric problems; there is no significant increase in domain/concept coverage between these increments of the ARC challenge, only the required adaptability and efficiency increases.

H.4 Generalization

There are indications that large, unspecific models are actually very good at generalization. Why then focus on the "model specificity" pillar? Researchers like Goldblum et al. [2024] have indeed observed that LLMs (or more generally, "overparameterized" neural networks) do not tend to overfit as much as originally thought but rather generalize over topics. However, they also claim that this effect is related to the low Kolmogorov complexity of real-world data. They therefore argue (and we agree) that currently, the *no free lunch* theorems have little relevance for SoTA LLMs. Acknowledging that LLMs work and generalize does not make them efficient at doing so. So this does not help us much for skill-acquisition efficiency. Datasets like ARC, which are precisely focused on skill acquisition efficiency, do not possess the same low complexity/flexibility as real-world data. As a consequence, custom under-specific ARC solvers have a much harder time performing on ARC tasks. We therefore think that *efficient generalization* indeed is scope-dependent, making the *no free lunch* theorems relevant again.

On that note: human intelligence is not universal either. We are relatively optimized/specialized for the specific physical world we live in. The concept of *universal general intelligence* exists, and humanity is pretty far away from it [Everitt and Hutter, 2018].